

SJ/B

42. (Added) The method of claim 41 wherein specifying the precedence information comprises selecting as the group of clients a rollout group of clients that is relatively larger than the pilot group and relatively smaller than the total number of clients of the network.

43. (Added) A computer-readable medium having computer-executable instructions for performing the method of claim 30.

REMARKS

The Office action dated July 17, 2002 ("the Office action") has been carefully considered. Claims 1-27 were pending in the application; of these claims 1-4, 6-7, 9, 14, 16-21, 26 and 27 were rejected under 35 U.S.C. 102(e) as being anticipated by Shrader et al., U.S. Patent No. 5,867,713 (hereinafter Shrader). Claims 5 and 11 (and also claim 25) were rejected under 35 U.S.C. 103(a) as being unpatentable over Shrader in view of Parthesarathy et al., U.S. Patent No. 6,269,480 (hereinafter Parthesarathy). Claims 8, 10, 13 and 15 were rejected under 35 U.S.C. 103(a) as being unpatentable over Shrader in view of Davis et al., U.S. Patent No. 5,742,289 (hereinafter Davis). Claims 12 and 22 were rejected under 35 U.S.C. 103(a) as being unpatentable over Shrader in view of Nakajima et al., U.S. Patent No. 6,289,510 (hereinafter Nakajima). Claims 23 and 24 were rejected under 35 U.S.C. 103(a) as being unpatentable over Shrader in view of Hendrickson et al., U.S. Patent No. 5,933,646 (hereinafter Hendrickson).

By the present amendment, claims 1-9, 11-18 and 20-27 have been amended, claims 28-43 have been added, and the rejections traversed in view of the following remarks. Applicants submit that the amendments herein are for purposes of clarifying the claims, and not for reasons related to patentability. Reconsideration is respectfully requested.

The present invention is generally directed to deploying software implementations, such as applications and other program components, when there is a potential or actual conflict among the software implementations that are specified (e.g., by administrators) to apply to a given user or machine. For example, network administrators may set policy objects to specify which programs and components apply to each user and machine, so that when a user logs on or a machine connects to the network, the user and machine automatically have appropriate programs and/or software components deployed thereto. Because different administrators can separately set the policies, and users and machines may belong to multiple, different groups to which policies apply, there may be a potential conflict among software implementations specified for deployment. By way of example, a user may belong to a finance group and an accounting group; if one policy specifies that the finance group is to use word processor X and another policy specifies that the accounting group is to use word processor Y, both applications may apply to that user. Rather than deploy both to a given machine to which the user has logged on, the present invention provides precedence information to handle the conflict, via defined precedence relationships between these applications, e.g., precedence data can specify to install word processor X and not word processor Y in the event both apply. Note that X and Y may be different versions of the same general program, and thus the present invention facilitates lifecycle management of software implementations. Further, note that the above summary is for example and informational purposes only and should not be used to interpret the claims, which are discussed below.

In contrast to the claims, Shrader is directed to a validation system that verifies an object-based installation plan for applications. Significantly, as part of validation, Shrader may have to add prerequisite child objects to an installation plan, but clearly does not teach or suggest making

any “selection” between which of two (or more) objects to add to the plan and which to not add, let alone do so based on anything even remotely resembling an established precedence relationship between such objects.

In fact, the very premise upon which the Office action has based its rejections, i.e., that Shrader’s child objects somehow involve precedence, is completely faulty and unsupported in Shrader, as well as being an unreasonable interpretation of the plain meaning of precedence. For example, the Office action has, (including on page 2 thereof), erroneously interpreted a precedence relationship to be disclosed by Shrader’s *required* files, or child objects. However, not only is such an interpretation clearly unreasonable given these words’ ordinary meanings, but this is directly opposite what applicants have described and claimed. In fact, if all files were required as in Shrader, then the very purpose of the present invention would be defeated, since the present invention is in part directed to *not* selecting certain software implementations (e.g., files) for deployment, and instead selecting others based on a precedence relationship. Clearly, any such non-selected files are not required. In fact, to the extent that Shrader’s child files are required (as suggested in Shrader and the Office action), then Shrader if anything would teach away from selection / selective deployment as generally recited in the claims of the present invention.

Further, although Shrader is mostly silent as to what the child objects that Shrader needs to add are actually added for, it is unquestionable that each are related to a single application (represented by a common parent object), and are not a plurality of (e.g., conflicting) software implementations that are both specified for deployment. There thus appears to be a general *dependence* relationship among files in Shrader (to add needed files), but nothing even remotely resembling a *precedence* relationship, let alone in the manner claimed, e.g., where precedence

information is used to select / deploy one specified software implementation and deselect (or not deploy) another specified software implementation.

By law, in order to support an anticipation rejection, the Office action must show that each and every element of the claimed invention is disclosed in a single reference, and that each element is arranged as in the claim. Shrader, which clearly does not teach or even suggest the subject matter of the independent claims, (which essentially recite maintaining and using precedence information to select / deploy one software implementation and deselect / not deploy another software implementation), is completely deficient in meeting these requirements as a matter of law. Reconsideration and withdrawal of the §102 rejections of claims 1-4, 6-7, 9, 14, 16-21, 27 and 27 based on Shrader is respectfully requested.

The rejections of the dependent claims also point out how the Office action has fundamentally misinterpreted Shrader's teachings with respect to the claimed subject matter. For example, claims 6 and 7 are directed to uninstalling one of the two software installations, as specified in the precedence data. However, in rejecting claim 6, the Office action has cited text in Shrader directed to removing the files and/or commands that are needed to facilitate plan execution, not the files that are installed via the plan. There is also no precedence involved in such a removal. The rejection of claim 9 is based on the Office action's faulty premise (discussed above) that Shrader's child objects have some type of precedence relationship, which is simply incorrect.

Regarding the rejections of claims 26 and 27, the Office action has cited column 2, lines 55-57, which are directed to customizing a *response* file for particular workstations. Such a customized response has nothing to do with specifying a pilot or rollout group of clients for

software implementation deployment. These rejections are completely unsupported by anything in Shrader.

Turning to the §103(a) rejections, applicants submit that the claims are patentable over Shrader and any of the other prior art of record for at least the reasons set forth above, e.g., like Shrader, none of the other prior art of record discloses, suggests or provides any motivation for handling the selective deployment of one software implementation over another, based on precedence information, when a plurality of software implementations are specified for deployment. In making these §103(a) rejections, applicants note that the Office action has again confused dependency with precedence, which are distinct concepts. Thus, even if somehow permissible to combine Shrader with Parthesarathy, Davis, Nakajima and/or Henderickson, the claims are not reached, as by law, in order to establish *prima facie* obviousness of a claimed invention, all of the claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974). In addition, “all words in a claim must be considered in judging the patentability of that claim against the prior art.” *In re Wilson*, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970).

Further, considering the §103(a) rejections as a whole, it appears that the Office has relied on applicants’ teachings in a (failed) attempt to reconstruct applicants’ invention. For example, instead of pointing out where some motivation to combine exists, the Office action merely makes conclusory statements based on applicants’ teachings; e.g., in rejecting claim 5, the Office action refers to “advertising” even though advertising as defined and claimed (as a software implementation available for execution prior to actual installation) is not considered in Parthesarathy, nor does it meet the definition used in the Office action. Similarly, in rejecting claim 11, the Office action again confuses dependency with precedence, and then refers to a

motivation to set precedence values during installation so as to configure programs correctly, which is wholly unrelated to the present invention (in the present invention, precedence data is used to determine which software implementation to install, and is not "set" during installation). Davis (column 14, lines 7-10) simply deletes files prior to an update; no precedence information is accessed. Nakajima (column 7, lines 5-10) is directed to tracking file installation states, not specifying what to do with files, let alone any precedence-based behavior. Henderickson refers to deinstallation, but is essentially silent as to how this is done, other than based on user specified changes.

For a combination of prior art references to render an invention obvious, there must be some reason, suggestion, or motivation found in the prior art whereby a person of ordinary skill in the field of the invention would make the combination. *In re Oetiker*, 977 F.2d 1443, 1447, 24 USPQ2d 1443, 1446 (Fed. Cir. 1992). A finding of obviousness on any other basis would constitute impermissible hindsight. See *Interconnect Planning Corp. v. Feil*, 774 F.2d 1132, 1143, 227 USPQ 543, 551 (Fed. Cir. 1985). Otherwise, combining prior art references without evidence of such a suggestion, teaching, or motivation simply takes the inventor's disclosure as a blueprint for piecing together the prior art to defeat patentability--the essence of impermissible hindsight. *In re Dembiczaik*, 175 F.3d 994, 999, 50 USPQ2d 1614, 1617 (Fed. Cir. 1999).

In the present application, the Office action has essentially done what is not proper by law, and used applicants' teachings as a blueprint, using an (incorrectly) modified Shrader for some of the claimed limitations, and, without any specific evidence of motivation to combine, has hunted for other references that might supply the limitations present in the application but missing from Shrader. Instead of presenting any specific evidence of motivation to combine, the Office action has only made conclusory statements that are wholly unrelated to the claims in

order to allege obviousness. However, such broad conclusory statements regarding the teaching of multiple references, standing alone, are not evidence of obviousness. *Id.*

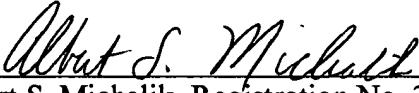
For at least these additional reasons, applicants submit that the claims are patentable over the prior art of record, whether considered alone or in any permissible combination, and respectfully request reconsideration and withdrawal of the rejections based thereon.

CONCLUSION

In view of the foregoing remarks, it is respectfully submitted that claims 1-43 of the present application should not be rejected on the art or otherwise, and that the application is good and proper form for allowance. A favorable action on the part of the Examiner is earnestly solicited.

If in the opinion of the Examiner a telephone conference would expedite the prosecution of the subject application, the Examiner is invited to call the undersigned attorney at (425) 836-3030.

Respectfully submitted,


Albert S. Michalik
Albert S. Michalik, Registration No. 37,395
Attorney for Applicants
Law Offices of Albert S. Michalik, PLLC
704 - 228th Avenue NE
Suite 193
Sammamish, WA 98074
(425) 836-3030 (telephone)
(425) 836-8957 (facsimile)

APPENDIX A

(Copy of amended claims showing additions and deletions made herein)

1. (Amended) In a network having software implementations deployed therein, a method for determining a set of software implementations to deploy to a client, comprising [the steps of], maintaining precedence information at a network location indicative of precedence relationships between software implementations, selecting a software implementation that is specified for deployment to the client as a selected software implementation, determining from the precedence information whether [if] the selected software implementation has precedence over at least one other software implementation that is also specified for deployment to the client, and if so, setting the selected software implementation for deployment and deselecting the at least one other software implementation.
2. (Amended) The method of claim 1 wherein [the step of] setting the selected software implementation for deployment comprises [the step of] setting the selected software implementation for install.
3. (Amended) The method of claim 2 wherein [the step of] setting the selected software implementation for install comprises [the step of] including the software implementation in a list of software implementations to install.
4. (Amended) The method of claim 1 further comprising [the step of] installing the selected software implementation.

5. (Amended) The method of claim 1 further comprising [the step of] advertising the selected software implementation as available to the user for execution on a computer system prior to actual installation of the software implementation on the computer system.

6. (Amended) The method of claim 1 wherein [the step of] deselecting comprises [the step of] setting the at least one other software implementation for uninstall.

7. (Amended) The method of claim 1 further comprising [the step of] uninstalling the at least one other software implementation.

8. (Amended) The method of claim 1 wherein [the step of] deselecting comprises [the step of] removing the at least one other software implementation from the set of software implementations to deploy.

9. (Amended) The method of claim 1 wherein the precedence information indicative of precedence relationships between software implementations is maintained in a [centralized] class store of the network.

11. (Amended) The method of claim 1 wherein the precedence information indicative of precedence relationships between software implementations is maintained in property values for the software implementations.

12. (Amended) The method of claim 1 wherein the precedence information indicative of precedence relationships between software implementations includes a property value indicative of whether to replace or overlay another software implementation.

13. (Amended) The method of claim 1 wherein the client is a user, and wherein [the step of] selecting a software implementation as a selected software implementation automatically occurs as part of a user logon.

14. (Amended) The method of claim 1 wherein the client is a user, and wherein [the step of] selecting a software implementation as a selected software implementation occurs in response to a user request.

15. (Amended) The method of claim 1 wherein the client is a machine, and wherein [the step of] selecting a software implementation as a selected software implementation automatically occurs when the machine connects to the network.

16. (Amended) A method for implementing a lifecycle for software implementations deployed in a network, comprising [the steps of], maintaining policy information at a network location including software implementations to deploy to groups of clients and precedence relationships between the software implementations, receiving [a] at least one request to deploy a plurality of software [implementation] implementations to a client of a particular group, locating the policy information corresponding to the group, and determining from the policy information which one of the plurality of software implementations to apply to

the client and which software implementation or implementations to not apply to the client based on the precedence relationships [maintained therefor].

17. (Amended) The method of claim 16 wherein [the step of] determining which one of the plurality of software implementations to apply to the client comprises [the step of] setting a first software implementation for install when the first software implementation has precedence over a second software implementation.

18. (Amended) The method of claim 17 further comprising [the step of] installing the first software implementation.

20. (Amended) The method of claim 18 wherein the client is a user, and wherein [the step of] installing the first software implementation is optional for that user.

21. (Amended) The method of claim 16 wherein [the step of] determining which one of the plurality of software implementations to apply to the client comprises [the step of] removing a first software implementation from a list of software implementations to install when a second software implementation has precedence over the first software implementation.

22. (Amended) The method of claim 21 further comprising [the step of] installing the second software implementation, wherein [the step of] installing the second software implementation overlays the first software implementation.

23. (Amended) The method of claim 16 wherein [the step of] determining which one of the plurality of software implementations to apply to the client comprises [the step of] setting a first software implementation for uninstall when a second software implementation has precedence over the second software implementation.

24. (Amended) The method of claim 23 further comprising [the step of] uninstalling the first software implementation and installing the second software implementation.

25. (Amended) The method of claim 16 further comprising [the step of] specifying the precedence relationships for the groups of clients.

26. (Amended) The method of claim 25 wherein [the step of] specifying the precedence relationships for the groups of clients comprises [the step of] specifying a pilot group of clients that is small relative to a total number of clients of the network.

27. (Amended) The method of claim 26 wherein [the step of] specifying the precedence relationships for the groups of clients comprises [the step of] specifying a rollout group of clients that is relatively larger than the pilot group and smaller than the total number of clients of the network.

CERTIFICATE OF MAILING

I hereby certify that this Amendment, Transmittal and Petition for Extension of Time are being deposited with the United States Postal Service on the date shown below, with sufficient postage as first class mail, in an envelope addressed to: Commissioner for Patents, Washington, D.C. 20231.

Date: January 16, 2003

Albert S. Michalik
Albert S. Michalik

1650 Amendment